

Practical 11

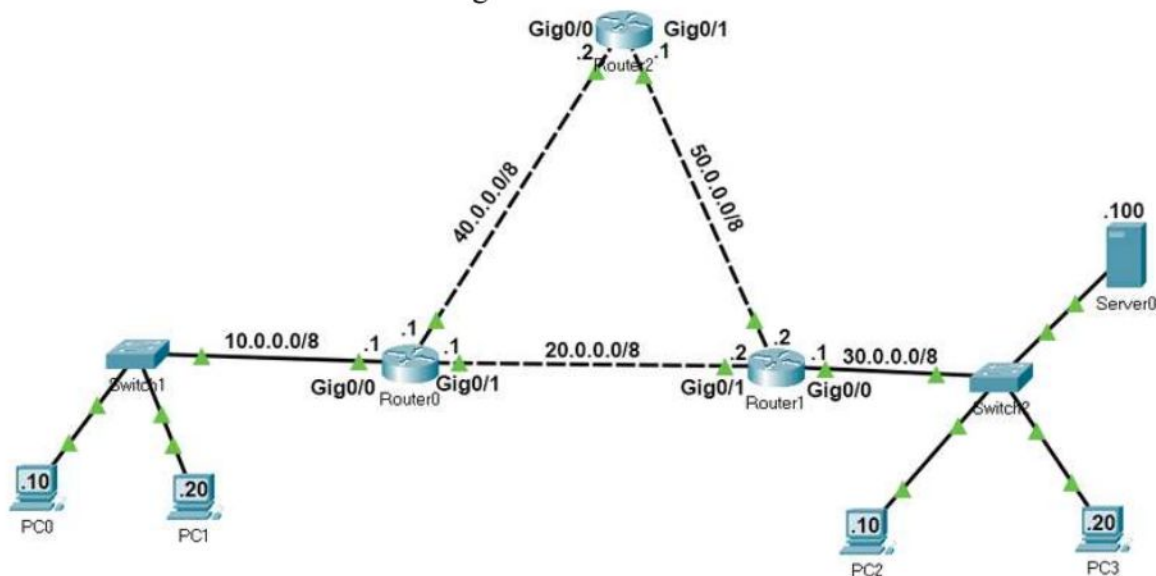
AIM:- a)Simulate Static Routing Configuration using CISCO Packet Tracer

Static routes are the routes you manually add to the router's routing table. The process of adding static routes to the routing table is known as static routing. Let's take a packet tracer example to understand how to use static routing to create and add a static route to the routing table.

Setting up a practice lab

Create a packet tracer lab as shown in the following image or download the following pre-created lab and load it on Packet Tracer.

Packet Tracer Lab with Initial IP Configuration



In this lab, each network has two routes to reach. We will configure one route as the main route and another route as the backup route. If the link bandwidth of all routes is the same, we use the route that has the least number of routers as the main route. If the link bandwidth and the number of routers are the same, we can use any route as the main route and another route as the backup route.

If we specify two routes for the same destination, the router automatically selects the best route for the destination and adds the route to the routing table. If you manually want to select a route that the router should add to the routing table, you have to set the AD value of the route lower than other routes. For example, if you use the following commands to create two static routes for network 30.0.0.0/8, the route will place the first route to the routing table.

```
#ip route 30.0.0.0 255.0.0.0 20.0.0.2 10
#ip route 30.0.0.0 255.0.0.0 40.0.0.2 20
```

If the first route fails, the router automatically adds the second route to the routing table.

Creating, adding, verifying static routes

Routers automatically learn their connected networks. We only need to add routes for the networks that are not available on the router's interfaces. For example, network 10.0.0.0/8, 20.0.0.0/8 and 40.0.0.0/8 are directly connected to Router0. Thus, we don't need to configure routes for these

networks. Network 30.0.0.0/8 and network 50.0.0.0/8 are not available on Router0. We have to create and add routes only for these networks.

The following table lists the connected networks of each router.

Router	Available networks on local interfaces	Networks available on other routers' interfaces
Router0	10.0.0.0/8, 20.0.0.0/8, 40.0.0.0/8	30.0.0.0/8, 50.0.0.0/8
Router1	20.0.0.0/8, 30.0.0.0/8, 50.0.0.0/8	10.0.0.0/8, 40.0.0.0/8
Router2	40.0.0.0/8, 50.0.0.0/8	10.0.0.0/8, 20.0.0.0/8, 30.0.0.0/8

Let's create static routes on each router for networks that are not available on the router.

Router0 requirements

- Create two routes for network 30.0.0.0/8 and configure the first route (via -Router1) as the main route and the second route (via-Router2) as a backup route.
- Create two routes for the host 30.0.0.100/8 and configure the first route (via -Router2) as the main route and the second route (via-Router1) as a backup route.
- Create two routes for network 50.0.0.0/8 and configure the first route (via -Router2) as the main route and the second route (via-Router1) as a backup route.
- Verify the router adds only main routes to the routing table.

Router0 configuration

Access the CLI prompt of Router0 and run the following commands.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2 10
Router(config)#ip route 30.0.0.0 255.0.0.0 40.0.0.2 20
Router(config)#ip route 30.0.0.100 255.255.255.255 40.0.0.2 10
Router(config)#ip route 30.0.0.100 255.255.255.255 20.0.0.2 20
Router(config)#ip route 50.0.0.0 255.0.0.0 40.0.0.2 10
Router(config)#ip route 50.0.0.0 255.0.0.0 20.0.0.2 20
Router(config)#exit
Router#show ip route static
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S 30.0.0.0/8 [10/0] via 20.0.0.2
S 30.0.0.100/32 [10/0] via 40.0.0.2
S 50.0.0.0/8 [10/0] via 40.0.0.2
Router#
```

```
Router0
Physical Config CLI Attributes
IOS Command Line Interface

Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 30.0.0.0 255.0.0.0 20.0.0.2 10 Primary route
Router(config)#ip route 30.0.0.0 255.0.0.0 40.0.0.2 20 Backup route
Router(config)#ip route 30.0.0.100 255.255.255.255 40.0.0.2 10 Primary route
Router(config)#ip route 30.0.0.100 255.255.255.255 20.0.0.2 20 Backup route
Router(config)#ip route 50.0.0.0 255.0.0.0 40.0.0.2 10 Primary route
Router(config)#ip route 50.0.0.0 255.0.0.0 20.0.0.2 20 Backup route
Router(config)#exit
Router#show ip route static
      30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S       30.0.0.0/8 [10/0] via 20.0.0.2
S       30.0.0.100/32 [10/0] via 40.0.0.2
S       50.0.0.0/8 [10/0] via 40.0.0.2
Router#
```

Router1 requirements

- Create two routes for network 10.0.0.0/8 and configure the first route (via -Router0) as the main route and the second route (via-Router1) as a backup route.
- Create two routes for network 40.0.0.0/8 and configure the first route (via -Router0) as the main route and the second route (via-Router2) as a backup route.
- Verify the router adds only main routes to the routing table.

Router1 configuration

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1 10
Router(config)#ip route 10.0.0.0 255.0.0.0 50.0.0.1 20
Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.1 10
Router(config)#ip route 40.0.0.0 255.0.0.0 50.0.0.1 20
Router(config)#exit
Router#show ip route static
S 10.0.0.0/8 [10/0] via 20.0.0.1
S 40.0.0.0/8 [10/0] via 20.0.0.1
Router#
```



```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 20.0.0.1 10 main route
Router(config)#ip route 10.0.0.0 255.0.0.0 50.0.0.1 20 backup route
Router(config)#ip route 40.0.0.0 255.0.0.0 20.0.0.1 10 main route
Router(config)#ip route 40.0.0.0 255.0.0.0 50.0.0.1 20 backup route
Router(config)#exit
Router#show ip route static
S    10.0.0.0/8 [10/0] via 20.0.0.1 } Only main routes are
S    40.0.0.0/8 [10/0] via 20.0.0.1 } added to the routing table.

Router#
```

Router2 requirements

Create static routes for network 10.0.0.0/8 and network 30.0.0.0/8 and verify the router adds both routes to the routing table.

Router2 configuration

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip route 10.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 30.0.0.0 255.0.0.0 50.0.0.2
Router(config)#exit
Router#show ip route static
S 10.0.0.0/8 [1/0] via 40.0.0.1
S 30.0.0.0/8 [1/0] via 50.0.0.2
Router#
```

 Router2

Physical Config CLI Attributes

IOS Command Line Interface

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with
Router(config)#ip route 10.0.0.0 255.0.0.0 40.0.0.1
Router(config)#ip route 30.0.0.0 255.0.0.0 50.0.0.2
Router(config)#exit
Router#show ip route static
S    10.0.0.0/8 [1/0] via 40.0.0.1
S    30.0.0.0/8 [1/0] via 50.0.0.2
Router#
```

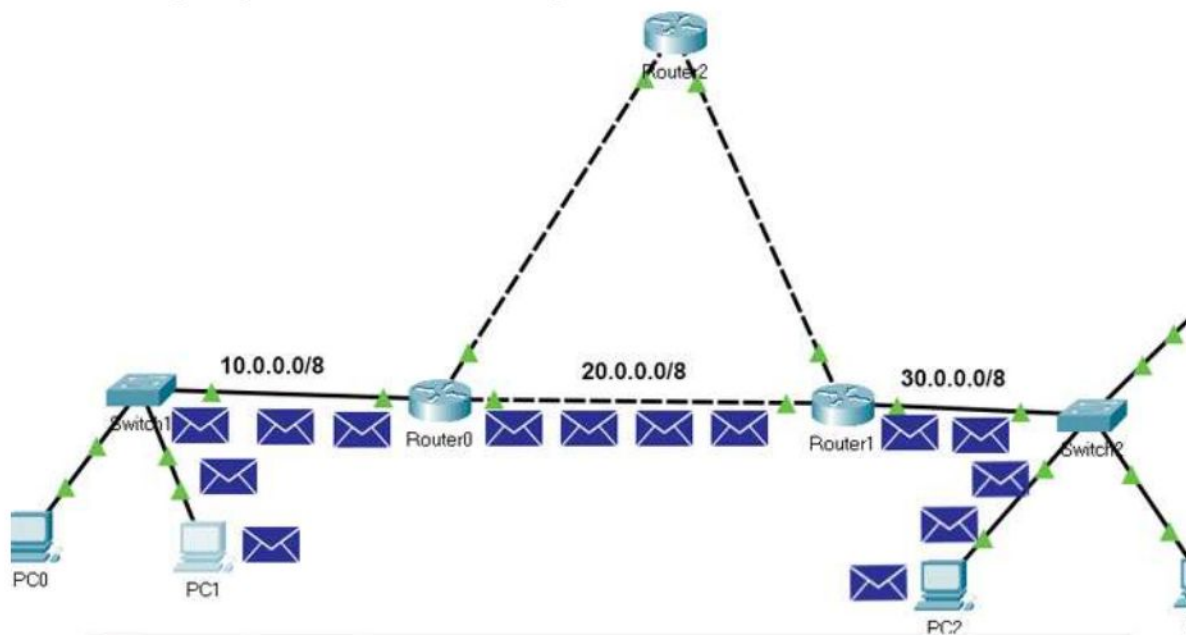
Verifying static routing

On Router0, we configured two routes for network 30.0.0.0/8. These routes are via Router1 and via Router2. We set the first route (via-Router1) as the main route and the second route as the backup route. We can verify this configuration in two ways.

By sending ping requests to a PC of network 30.0.0.0/8 and tracing the path they take to reach the network 30.0.0.0/8. For this, you can use '**tracert**' command on a PC of network 10.0.0.0/8. The '**tracert**' command sends ping requests to the destination host and tracks the path they take to reach the destination.

By listing the routing table entries on Router0. Since a router uses the routing table to forward data packets, you can check the routing table to figure out the route the router uses to forward data packets for each destination.

The following image shows the above testing.



```
Router0
Physical Config CLI Attributes
IOS Command Line Interface

Router>enable
Router#show ip route static
    30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S    30.0.0.0/8 [10/0] via 20.0.0.2
S    30.0.0.100/32 [10/0] via 40.0.0.2
S    50.0.0.0/8 [10/0] via 40.0.0.2
Router#
```

```
PC1
Physical Config Desktop Programming Attributes
Command Prompt

C:\>tracert 30.0.0.20

Tracing route to 30.0.0.20 over a maximum of 30
hops:

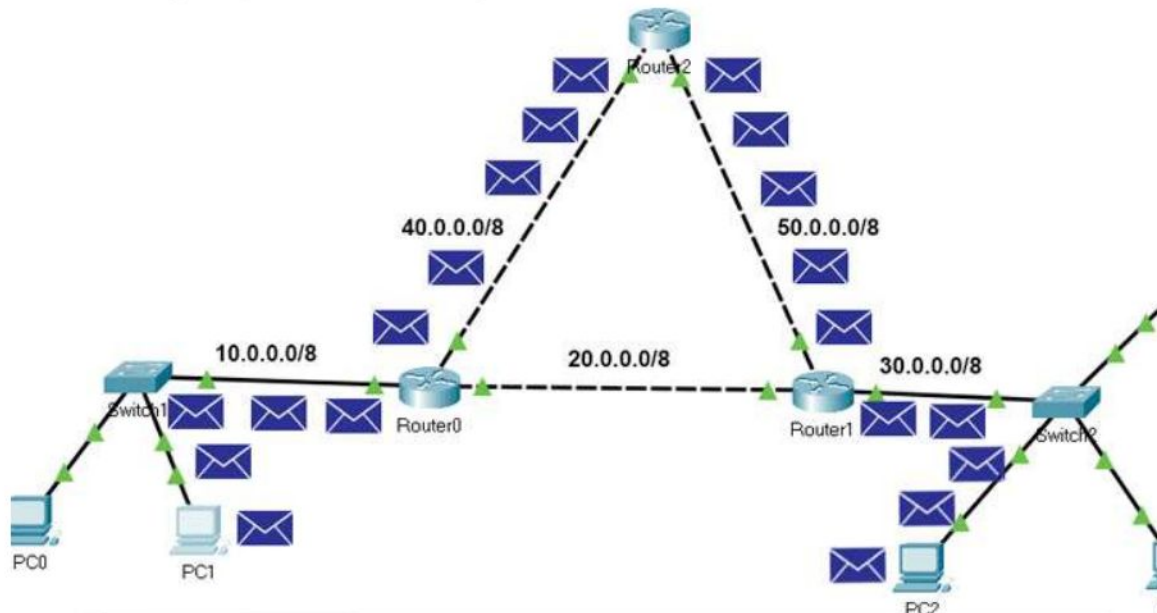
  1  0 ms      0 ms      0 ms      10.0.0.1
  2  0 ms      0 ms      0 ms      20.0.0.2
  3  *         0 ms      0 ms      30.0.0.20

Trace complete.

C:\>
```

We also configured a separate static host route for the host 30.0.0.100/8. The router must use this route to forward data packets to the host 30.0.0.100/8. To verify this, you can do the same testing for the host 30.0.0.100/8.

The following image shows this testing.



```
Router0
Physical Config CLI Attributes
IOS Command Line Interface

Router>enable
Router#show ip route static
      30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S       30.0.0.0/8 [10/0] via 20.0.0.2
S       30.0.0.100/32 [10/0] via 40.0.0.2
S       50.0.0.0/8 [10/0] via 40.0.0.2

Router#
```

```
PC1
Physical Config Desktop Programming Attributes

Packet Tracer PC Command Line 1.0
C:\>tracert 30.0.0.100

Tracing route to 30.0.0.100 over a maximum of 30 hops:

  1  1 ms    0 ms    0 ms    10.0.0.1
  2  *       0 ms    0 ms    40.0.0.2
  3  *       *      0 ms    20.0.0.2
  4  *       0 ms    0 ms    30.0.0.100

Trace complete.
C:\>
```


We also configured a backup route for network 30.0.0.0/8. The router must put the backup route to the routing table and use it to forward data packets to network 30.0.0.0/8 when the main route fails. To verify this, we have to simulate the failure of the main route.

To simulate the failure of the main route, you can delete the link between Router0 and Router1. After deleting the link, do the same testing again for the network 30.0.0.0/8.

The network diagram shows a topology with three routers: Router0, Router1, and Router2. Router0 is connected to Router1, and Router1 is connected to Router2. Router0 is also connected to Switch1, which is connected to PC0 and PC1. Router1 is connected to Switch2, which is connected to Server0. The link between Router0 and Router1 is highlighted with a red 'X' and labeled 'If the main route is up.'.

The screenshots show the following commands and outputs:

```

C:\>tracert 30.0.0.20

Tracing route to 30.0.0.20 over a maximum of 30 hops:
  0  0 ms  0 ms  0 ms  10.0.0.1
  1  0 ms  0 ms  0 ms  20.0.0.2
  2  *    0 ms  0 ms  30.0.0.20
Trace complete.

C:\>tracert 30.0.0.20

Tracing route to 30.0.0.20 over a maximum of 30 hops:
  0  0 ms  0 ms  0 ms  10.0.0.1
  1  0 ms  0 ms  0 ms  40.0.0.2
  2  0 ms  0 ms  0 ms  50.0.0.2
  3  0 ms  0 ms  0 ms  30.0.0.20
Trace complete.
C:\>
  
```

The Router CLI shows the following commands and outputs:

```

Router>enable
Router#show ip route static
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S    30.0.0.0/8 [10/0] via 20.0.0.2
S    30.0.0.100/32 [10/0] via 40.0.0.2
S    50.0.0.0/8 [10/0] via 40.0.0.2

Router#
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to down

Router#show ip route static
30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S    30.0.0.0/8 [20/0] via 40.0.0.2
S    30.0.0.100/32 [10/0] via 40.0.0.2
S    50.0.0.0/8 [10/0] via 40.0.0.2

Router#
  
```

The output shows that when the main route is up, the traffic goes through the main route (10.0.0.1 to 20.0.0.2 to 30.0.0.20). When the main route is down, the traffic goes through the backup route (10.0.0.1 to 40.0.0.2 to 50.0.0.2 to 30.0.0.20).

The following link provides the configured packet tracer lab of the above example.
[Packet Tracer Lab with Static Routing Configuration](#)

Deleting a static route

To delete a static route, use the following steps.

- Use the '**show ip route static**' command to print all static routes.
- Note down the route you want to delete.
- Use the '**no ip route**' command to delete the route.

If you have a backup route, the backup route becomes the main route when you delete the main route.

In our example, we have a backup route and a main route for the host 30.0.0.100/8. The following image shows how to delete both routes.

Router0

Physical Config CLI Attributes

IOS Command Line Interface

```
Router>enable
Router#show ip route static
    30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S       30.0.0.0/8 [10/0] via 20.0.0.2
S       30.0.0.100/32 [10/0] via 40.0.0.2
S       50.0.0.0/8 [10/0] via 40.0.0.2

Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip route 30.0.0.100 255.255.255.255 40.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route static
    30.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S       30.0.0.0/8 [10/0] via 20.0.0.2
S       30.0.0.100/32 [20/0] via 20.0.0.2
S       50.0.0.0/8 [10/0] via 40.0.0.2

Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no ip route 30.0.0.100 255.255.255.255 20.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#show ip route static
S       30.0.0.0/8 [10/0] via 20.0.0.2
S       50.0.0.0/8 [10/0] via 40.0.0.2
Router#
```

The main route that we want to delete.

Deleting the main route

As soon as we remove the main route, the router changes the backup route to the main route.

Deleting the new main route

All routes to host 30.0.0.100/8 have been removed.

Practical 11

AIM:- b)Simulate RIP using CISCO Packet Tracer

Initial IP configuration

Device	Interface	IP Configuration	Connected with
PC0	Fast Ethernet	10.0.0.2/8	Router0's Fa0/1
Router0	Fa0/1	10.0.0.1/8	PC0's Fast Ethernet
Router0	S0/0/1	192.168.1.254/30	Router2's S0/0/1
Router0	S0/0/0	192.168.1.249/30	Router1's S0/0/0
Router1	S0/0/0	192.168.1.250/30	Router0's S0/0/0
Router1	S0/0/1	192.168.1.246/30	Router2's S0/0/0
Router2	S0/0/0	192.168.1.245/30	Router1's S0/0/1
Router2	S0/0/1	192.168.1.253/30	Router0's S0/0/1
Router2	Fa0/1	20.0.0.1/30	PC1's Fast Ethernet
PC1	Fast Ethernet	20.0.0.2/30	Router2's Fa0/1

Assign IP address to PCs

Double click **PCs** and click **Desktop** menu item and click **IP Configuration**. Assign IP address referring the above table.

Assign IP address to interfaces of routers

Double click **Router0** and click **CLI** and press **Enter** key to access the command prompt of **Router0**.

We need to configure IP address and other parameters on interfaces before we could actually use them for routing. Interface mode is used to assign IP address and other parameters. Interface mode can be accessed from global configuration mode. Following commands are used to access the global configuration mode.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

From global configuration mode we can enter in interface mode. From there we can configure the interface. Following commands will assign IP address on FastEthernet0/0.

```
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

interface fastEthernet 0/0 command is used to enter in interface mode.

ip address 10.0.0.1 255.0.0.0 command will assign IP address to interface.

no shutdown command will bring the interface up.

exit command is used to return in global configuration mode.

Serial interface needs two additional parameters **clock rate** and **bandwidth**. Every serial cable has two ends DTE and DCE. These parameters are always configured at DCE end.

We can use **show controllers interface** command from privilege mode to check the cable's end.

```
Router#show controllers serial 0/0/0
Interface Serial0/0/0
Hardware is PowerQUICC MPC860
DCE V.35, clock rate 2000000
[Output omitted]
```

Fourth line of output confirms that DCE end of serial cable is attached. If you see DTE here instead of DCE skip these parameters.

Now we have necessary information let's assign IP address to serial interface.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.1.249 255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.1.254 255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#
```

Router#configure terminal Command is used to enter in global configuration mode.

Router(config)#interface serial 0/0/0 Command is used to enter in interface mode.

Router(config-if)#ip address 192.168.1.249 255.255.255.252 Command assigns IP address to interface. For serial link we usually use IP address from /30 subnet.

Router(config-if)#clock rate 64000 And **Router(config-if)#bandwidth 64** In real life environment these parameters control the data flow between serial links and need to be set at service providers end. In lab environment we need not to worry about these values. We can use these values.

Router(config-if)#no shutdown Command brings interface up.

Router(config-if)#exit Command is used to return in global configuration mode.

We will use same commands to assign IP addresses on interfaces of remaining routers. We need to provide clock rate and bandwidth only on DCE side of serial interface. Following command will assign IP addresses on interface of Router1.

Router1

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.1.250 255.255.255.252
```



```
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.1.246 255.255.255.252
Router(config-if)#clock rate 64000
Router(config-if)#bandwidth 64
Router(config-if)#no shutdown
Router(config-if)#exit
```

Use same commands to assign IP addresses on interfaces of Router2.

Router2

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface fastEthernet 0/0
Router(config-if)#ip address 20.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/0
Router(config-if)#ip address 192.168.1.245 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface serial 0/0/1
Router(config-if)#ip address 192.168.1.253 255.255.255.252
Router(config-if)#no shutdown
Router(config-if)#exit
```

Now routers have information about the networks that they have on their own interfaces. Routers will not exchange this information between them on their own. We need to implement RIP routing protocol that will insist them to share this information.

Configure RIP routing protocol

Configuration of RIP protocol is much easier than you think. It requires only two steps to configure the RIP routing.

- Enable RIP routing protocol from global configuration mode.
- Tell RIP routing protocol which networks you want to advertise.

Let's configure it in Router0

Router0

```
Router0(config)#router rip
Router0(config-router)# network 10.0.0.0
Router0(config-router)# network 192.168.1.252
Router0(config-router)# network 192.168.1.248
```

router rip command tell router to enable the RIP routing protocol.

network command allows us to specify the networks which we want to advertise. We only need to specify the networks which are directly connected with the router.

That's all we need to configure the RIP. Follow same steps on remaining routers.

Router1

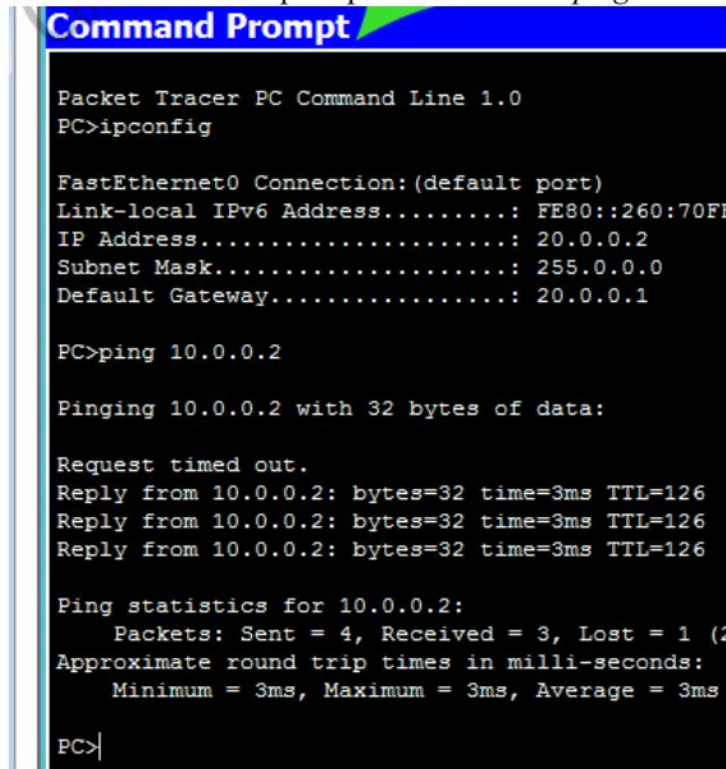
```
Router1(config)#router rip
Router1(config-router)# network 192.168.1.244
Router1(config-router)# network 192.168.1.248
```

Router2

```
Router2(config)#router rip
```

```
Router2(config-router)# network 20.0.0.0
Router2(config-router)# network 192.168.1.252
Router2(config-router)# network 192.168.1.244
```

That's it. Our network is ready to take the advantage of RIP routing. To verify the setup we will use ping command. ping command is used to test the connectivity between two devices. Access the command prompt of **PC1** and use *ping* command to test the connectivity from **PC0**.



```
Command Prompt
Packet Tracer PC Command Line 1.0
PC>ipconfig

FastEthernet0 Connection:(default port)
Link-local IPv6 Address.....: FE80::260:70FB
IP Address.....: 20.0.0.2
Subnet Mask.....: 255.0.0.0
Default Gateway.....: 20.0.0.1

PC>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 10.0.0.2: bytes=32 time=3ms TTL=126
Reply from 10.0.0.2: bytes=32 time=3ms TTL=126
Reply from 10.0.0.2: bytes=32 time=3ms TTL=126

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25%)
    Approximate round trip times in milli-seconds:
        Minimum = 3ms, Maximum = 3ms, Average = 3ms

PC>|
```

RIP protocol automatically manage all routes for us. If one route goes down, it automatically switches to another available. To explain this process more clearly we have added one more route in our network.

Currently there are two routes between PC0 and PC1.

Route 1

PC0 [Source / destination – 10.0.0.2] <==> Router0 [FastEthernet0/1 – 10.0.0.1] <==> Router0 [Serial0/0/1 – 192.168.1.254] <==> Router2 [Serial 0/0/1 – 192.168.1.253] <==> Router2 [FastEthernet0/0 – 20.0.0.1] <==> PC1 [Destination /source – 20.0.0.2]

Route 2

PC0 [Source / destination – 10.0.0.2] <==> Router0 [FastEthernet0/1 – 10.0.0.1] <==> Router0 [Serial0/0/0 – 192.168.1.249] <==> Router1 [Serial 0/0/0 – 192.168.1.250] <==> Router1 [Serial 0/0/1 – 192.168.1.246] <==> Router2 [Serial 0/0/0 – 192.168.1.245] <==> Router2 [FastEthernet0/0 – 20.0.0.1] <==> PC1 [Destination /source – 20.0.0.2]

By default RIP will use the route that has low hops counts between source and destination. In our network route1 has low hops counts, so it will be selected. We can use *tracert* command to verify it.

Now suppose route1 is down. We can simulate this situation by removing the cable attached between **Router0 [s0/0/1]** and **Router2 [s0/0/1]**.

What will happen now? There is no need to worry. RIP will automatically reroute the traffic. Use *tracert* command again to see the magic of dynamic routing.